Subject: Re: Network namespaces a path to mergable code.
Posted by ebiederm on Wed, 28 Jun 2006 16:51:26 GMT
View Forum Message <> Reply to Message

Andrey Savochkin <saw@swsoft.com> writes:

> Ok, fine.
> Now I'm working on socket code.
>
> We still have a question about implicit vs explicit function parameters.
> This question becomes more important for sockets: if we want to allow to use
> sockets belonging to namespaces other than the current one, we need to do
> something about it.

There seems to be some real benefit to that. Especially for things like NFS,
that captures the context at mount time. It might as well keep the namespace
in it's socket.

> One possible option to resolve this question is to show 2 relatively short
> patches just introducing namespaces for sockets in 2 ways: with explicit
> function parameters and using implicit current context.
> Then people can compare them and vote.
> Do you think it's worth the effort?

Given that we have two strong opinions in different directions I think it
is worth the effort to resolve this.

In a slightly different vein your second patch introduced a lot
of #ifdef CONFIG_NET_NS in C files. That is something we need to look closely
at.

So I think the abstraction that we use to access per network namespace
variables needs some work if we are going to allow the ability to compile
out all of the namespace code. The explicit versus implicit lookup is just
one dimension of that problem.

>> All minor things. The typo I was referring to was a section where the
>> original iteration was on an ifp variable and you called it dev
>> without changing the rest of the code in that section.
>>
>> The only big issue was that the patch too big, and should be split
>> into a patchset for better review. One patch for the new functions,
>> and the an additional patch for each driver/subsystem hunk describing
>> why that chunk needed to be changed.
>
> I'll split the patch.

Thanks.

>> I'm still curious why many of those chunks can't use existing helper
>> functions, to be cleaned up.
>
> What helper functions are you referring to?

Basically most of the device list walker functions live in.
net/core/dev.c

I don't know if the cases you fixed could have used any of those
helper functions but it certainly has me asking that question.

A general pattern that happens in cleanups is the discovery
that code using an old interface in a problematic way really
could be done much better another way. I didn't dig enough
to see if that was the case in any of the code that you changed.

Eric
